Niveau(x) d'enseignement: 0 Numéro EDUQ : 0470159
Centre d'intérêt: 640

Mataigne, Bernard.

Outils d'auto-évaluation des didacticiels en cours de production / Bernard Mataigne avec la collab. de Claude Séguin ; [pour la Direction de la technologie éducative] Service Orientations et développement [et la Direction de la production et la distribution du matériel didactique]. -- [Québec] : Direction générale des moyens d'enseignement, 1982. 36 p. en pag. multiple ; 28 cm.

La couv. porte en outre: Didacticiels. Code: 54-9626.

Descripteur(s): *Didacticiel *Evaluation du matériel didactique *Guide *Questionnaire

Identificateur(s) de sujet: *Elaboration de matériel didactique

Ce guide constitue un outil de soutien aux équipes et aux individus produisant du matériel didactique informatisé. Son objectif est de contribuer à ce que les didacticiels répondent davantage à des critères d'efficience et à ce qu'ils respectent les normes en vigueur dans le domaine des matériels didactiques. L'évaluation de la qualité des didacticiels est abordée sous quatre aspects: l'utilisation du français, l'interaction étudiant-ordinateur, le respect du système international d'unité et la transférabilité. Chacun de ces thèmes est analysé en trois étapes: la conception, le codage et l'expérimentation. La condition de passage d'une étape à l'autre est de répondre adéquatement aux questions posées à la fin de chaque étape.

Type de document: d

EDUQ 8470159

Direction de la technologie éducative
Direction de la production et de la distribution
du matériel didactique

OUTILS D'AUTO-ÉVALUATION DES DIDACTICIELS EN COURS DE PRODUCTION

par Bernard Mataigne
avec la collaboration de Claude Séguin

Gouvernement du Québec
Ministère de l'Éducation
Direction générale
des moyens d'enseignement

SERVICE ORIENTATIONS ET DÉVELOPPEMENT

OUTILS D'AUTO-ÉVALUATION DES DIDACTICIELS EN COURS DE PRODUCTION

Bernard Mataigne
avec la collaboration de
Claude Séguin

CODE: 84-9626
Novembre 1982

TABLE DES MATIERES

Avertissement ..... p. I
Introduction ..... p. II
Interaction étudiant-ordinateur ..... p. 1.1.1

Etape design ..... p. 1.1.1
Etape codage ..... p. 1.2.1
Etape expérimentation ..... p. 1.3.1

Respect et utilisation du français ..... p. 2.1.1

Etape design ..... p. 2.1.1
Etape codage ..... p. 2.2.1
Etape expérimentation ..... p. 2.3.1

Système international d'unités ..... p. 3.1.1

Etape design ..... p. 3.1.1
Etape codage ..... p. 3.2.1
Etape expérimentation ..... p. 3.3.1

Transférabilité ..... p. 4.1.1

Etape design ..... p. 4.1.1
Etape codage ..... p. 4.2.1
Etape expérimentation ..... p. 4.3.1

Notes de discussion ..... p. A.1

AVERTISSEMENT

Le document est le fruit de la collaboration de la Direction de la technologie éducative et de la Direction de la production et de la distribution du matériel didactique dans le cadre d'un soutien à la production de matériel didactique informatisé au niveau collégial. Bernard Mataigne (de la D.T.E.) a conçu, rédigé et revu le document; Claude Séguin (de la D.P.D.M.D.) a collaboré étroitement à sa conception, sa rédaction et sa révision.

Il contribue aussi au développement d'instruments d'évaluation mis à la disposition de tous les utilisateurs de l'informatique à des fins pédagogiques.

INTRODUCTION

Ce document se veut un outil de soutien aux équipes et aux individus produisant du matériel didactique informatisé. Ce document aborde quatre aspects de la qualité d'un tel matériel: l'interaction étudiant-ordinateur, l'utilisation du français, le respect du système international d'unité et la transférabilité. Ainsi nous souhaitons que les productions répondront davantage à des critères d'efficience et de «solidité» et que par ailleurs elles respecteront des normes admises dans tout matériel didactique.

Chacun des thèmes abordé est analysé en trois étapes: la conception, le codage et l'expérimentation (1). La condition de passage d'une étape à une autre est de répondre adéquatement aux questions posées à la fin de chaque étape. Ces questions synthèses, ici regroupées en fin de document sous le titre «Notes de discussions», sont des points de contrôle; ils doivent donc être appliqués avec intelligence et souplesse, en tenant compte du texte qui les précède et du contenu spécifique dont sera fait le produit final.

D'autres thèmes sont implicitement présents au texte sans être l'objet d'un regard particulier: la documentation, l'esthétique de la présentation... D'autres thèmes ne sont pas abordés ici mais pourraient l'être dans une refonte de ce document. On pourrait alors élargir le champ de l'évaluation à des préoccupations comme la pertinence didactique et l'intégration à la situation d'enseignement.

---

(1) Conception: étape d'analyse, de conceptualisation et de préparation du programme (j'utilise aussi «designe»).

Codage: étape de mise en forme concrète du programme à des fins d'exécution.

Expérimentation: étape de mise-en-pratique expérimentale du programme auprès du groupe d'étudiants.

Ce document est donc un instrument de travail suffisamment général pour qu'on puisse l'appliquer à différents types de produits (exerciseurs, tutoriels...) mais suffisamment exigeant pour que les équipes assurent un auto-contrôle qualitatif du produit.

ÉVALUATION

INTERACTION ÉTUDIANT-ORDINATEUR

ÉTAPE DESIGN

Il faudra étudier deux dimensions du rapport étudiant-ordinateur:

- ce que dit l'ordinateur (instructions, etc.).
- ce que peut dire l'étudiant à l'ordinateur.

A l'étape design, on examinera la conception générale du point de vue du déroulement du programme et de la structure prévue pour la manipulation des modules de programmation d'entrée et d'affichage.

1. Déroulement du programme

- comment le programme se déroule-t-il?
  - de façon linéaire?
  - en arborescence?
  - selon l'utilisateur?
- existe-t-il un module de présentation ou d'introduction intégré au didacticiel:
  - explication du programme, des objectifs visés, de la méthode utilisée?
  - prévoit-on une documentation imprimée pour compléter les explications?
- existe-t-il des modules de transition pour le passage d'une étape à

une autre au sein du programme? Ex: comment passe-t-on d'un «chapitre» à un autre, par choix dans un menu, automatiquement...?

- comment prévoit-on la fin du programme?

  - arrêt brutal de l'ordinateur?
  - récapitulation?
  - message avertissant de la fin du programme?
  - évaluation (note, etc.)?
  - retour au début...
  - impression des résultats?

Du point de vue de l'étudiant, on se demandera s'il est prévu des mécanismes de manipulation des "inputs" de l'étudiant:

  - analyseur de réponse (validation syntaxique et sémantique)
  - suggestions, rappels, conseils selon la réponse donnée
  - module de secours pour dépanner l'utilisateur (ex: commande S.O.S.).

On se demandera aussi si l'étudiant peut ou non modifier le déroulement du programme? Si oui, comment?

De plus, prévoit-on ce qui arrivera si l'étudiant fait une erreur (syntaxique): ex: la lettre «L» au lieu du chiffre «1»:

  - le programme s'arrête-t-il?
  - faut-il relancer tout le programme?
  - y a-t-il un module de récupération? (message, etc.)?
  - l'étudiant peut-il se reprendre?

2. <u>Affichage</u>

Ici, on étudiera comment sont traités le texte, le graphique et le son.

Le texte: respect et utilisation du français (voir la fiche d'évaluation).

Le graphique: est-il traité par modules de programmation distincts?

Le son: est-il traité par modules distincts?

Plus globalement, il faudra examiner l'efficacité générale du design de façon à ce que les temps de réponse soient acceptables pour l'utilisateur (de l'ordre des 2 secondes au maximum). Si un temps de réponse est supérieur à 2 secondes, il faudra prévenir l'utilisateur.

RÉSUMÉ

A l'étape design, il faudra donc se demander:

- le programme répondra-t-il à des critères efficaces d'interaction (modularité, simplicité, interactivité, souplesse)?

oui ◯ non ◯

sinon, on cherchera à épurer et à simplifier la démarche générale.

- le programme permettra-t-il le maximum de contrôle par l'étudiant (selon les finalités)?

oui ◯ non ◯

Sinon, on cherchera à assurer à l'utilisateur le plus grand contrôle possible sur la machine et le déroulement en respectant les objectifs didactiques qu'on poursuit dans le programme.

Si on répond oui aux deux questions, il y a passage à l'étape codage.

ETAPE CODAGE

A cette étape, la première vérification consistera à confronter le programme codé au design établi précédemment: la structure générale et les modules spécifiques sont-ils réalisés?

1. Plus précisément, on étudiera le rapport des divers modules entre eux, comment se fait le passage de l'un à l'autre, la situation des modules au sein du programme, la façon dont le programme les utilise (séquence, fréquence...). En fait, on verra à l'application de critères d'efficience en programmation. Cette efficacité sera fonction des finalités du programme. On pourra, par exemple, utiliser des critères tels que programmation structurée, rapidité, lisibilité, fiabilité, économie d'espace, transférabilité...

Du côté de la présentation et des instructions, on évaluera la qualité des textes selon les critères établis dans la fiche "Respect et utilisation du français". On surveillera plus particulièrement:

- la qualité du texte "parlant" à l'étudiant: (près du langage parlé, familier mais non vulgaire, personnalisé, simple...)

- la quantité de texte et d'images affichés (une page-écran ne doit pas dépasser une certaine "charge" d'information. Cette charge se situant autour de notre capacité à traiter, puis à mémoriser immédiatement des informations nouvelles).

- l'esthétique de la présentation: rapport texte/graphique, utilisation des couleurs, des sons, des soulignés, clignotants et autres techniques. On retiendra les critères suivants:

  : simplicité
  : homogénéité
  : souplesse

Le codage, à ce niveau, devra donc être conçu en terme de page-écran. Chaque programme doit donc comporter minimalement, en plus des pages-contenu:

- une page titre avec identification des produits et des producteurs
- une page de présentation-explication-instruction
- une page titre de chapitre (s'il y a lieu)
- une page "fin" (temporaire ou définitive).

On s'assurera aussi que le codage de la présentation graphique (sinon sonore) soit fait dans des modules autonomes, distincts de l'ensemble. On garantira ainsi une transférabilité plus facile.

2. Par rapport aux intrants de l'étudiant, on vérifiera:

a) la forme de ces intrants
b) la façon dont ils sont traités
c) le blindage par rapport à ceux-ci.

a) Le codage doit simplifier au maximum le rapport avec la machine. Il faut donc que la forme dans laquelle l'étudiant doit "entrer" ses informations soit simple, claire et facile:

- s'il s'agit de l'utilisation d'un clavier, les intrants ne seront pas trop longs et préférablement uniformes.
- s'il s'agit de périphériques autres, les instructions devront être claires et précises.

Rappelons ici que la machine doit être «transparente pour l'utilisateur.»

b) On vérifiera la façon dont sont traités les intrants: analyseur de réponse, etc. Une telle évaluation se fera en fonction des finalités mêmes du programme.

Si le programme comptabilise les intrants (bonnes réponses par exemple), renvoie des conseils ou suggestions, on verra à ce que ces procédures soient expliquées à l'étudiant (à l'intérieur du programme ou par documentation externe).

c) Enfin, le codage protégera le programme contre une utilisation naïve ou des erreurs non prévues (mauvais intrants, "break" ou "reset" inopinés, intrant vide, majuscule/minuscule, espace, etc.). Il faut donc prévoir des mécanismes de protection et/ou de récupération.

RÉSUMÉ

- Le codage respecte-t-il le design établi (modules, affichages...)?

oui ◯ non ◯

- L'utilisateur dispose-t-il d'une bonne documentation du programme tant à l'affichage que par documentation externe?

oui ◯ non ◯

- Le codage assure-t-il un maximum de facilité d'utilisation pour l'étudiant?

oui ◯ non ◯

- Le codage assure-t-il une bonne esthétique de chaque page-écran?

oui ◯ non ◯

- Le codage protège-t-il le programme au maximum?

oui ◯ non ◯

Dans le cas où on répondrait non à l'une ou à plusieurs des questions précédentes, il faudra reviser le codage en tenant compte des critères et indices établis plus avant.

Dans le cas contraire, on passera à l'étape expérimentation.

ETAPE EXPERIMENTATION

Il va de soi que l'expérimentation de la relation interactive étudiant-ordinateur ne peut être dissociée du contenu même du produit. L'évaluation de l'aspect «transmission de connaissances» n'est pourtant pas notre objet ici: le producteur doit assurer la meilleure dimension pédagogique de son didacticiel en collaboration avec les autres intervenants (D.G.M.E., étudiants, collègues).

Ici, nous nous attacherons donc surtout aux aspects «périphériques» à l'apprentissage comme tel.

Il faudra distinguer les problèmes liés à l'interaction ordinateur-étudiant de ceux liés à:

- une première utilisation de l'ordinateur
- une méconnaissance du contenu du programme (matière à étudier ou à expérimenter).

On demandera donc: est-ce la première fois que l'étudiant utilise l'ordinateur?

Par ailleurs, il sera souhaitable d'interroger les étudiants-utilisateurs sur les aspects suivants:

- le principal problème d'utilisation rencontré?
- la clarté des instructions? Sont-elles facilement compréhensibles?
- la "lisibilité" des pages-écrans? Est-ce facile de les comprendre? Les graphiques, le son et la couleur sont-ils bien utilisés?
- l'utilisation des périphériques d'entrée? Est-il facile d'utiliser le clavier (pas de trop longues

utilisations)? Est-il facile de manipuler les autres périphériques?

- l'étudiant a-t-il eu souvent recours à la documentation ou à l'aide du professeur (appariteur) pour avancer dans le programme?

- le contrôle sur le programme? L'étudiant a-t-il la sensation d'avoir un certain contrôle sur le déroulement ou, au contraire, se sent-il dirigé par celui-ci?

- les temps de réponse sont-ils satisfaisants ou trop longs? S'il faut attendre, l'étudiant sait-il ce qui se passe?

Il sera souhaitable de non seulement poser des questions à l'utilisateur mais de l'observer en cours d'expérimentation afin de pouvoir relever les indices les plus flagrants de problèmes.

Par ailleurs, l'enseignant (s'il n'est pas l'auteur), indiquera les principales difficultés d'interfaçage telles qu'il les a perçues à l'expérimentation: questions des étudiants, problèmes intervenus...

D'autre part, ce professeur (appariteur) a-t-il été en mesure de répondre aux problèmes et questions soulevés? La documentation du professeur fut-elle suffisante? Le programme exige-t-il de l'enseignant des connaissances particulières en informatique?

RÉSUMÉ

Existe-t-il des problèmes sérieux et/ou systématiques dans le rapport étudiant-programme?

oui ◯ non ◯

Si oui, renvoyer les commentaires pertinents à l'auteur.

ÉVALUATION

RESPECT ET UTILISATION DU FRANÇAIS

ÉTAPE DESIGN

Ici, on s'interrogera sur la pertinence et la nécessité de modules de programmation questions-réponses (ou même commentaires) qui favoriseront une homogénéité du rapport interactif. Ainsi, on favorisera des questions-réponses mettant en jeu des mots (oui/non) français plutôt que des symboles ou des chiffres (0/1). On favorisera aussi l'utilisation des formules légèrement variées mais d'un niveau homogène quand il faudra commenter ou "apprécier" des réponses ou tout simplement "parler" à l'usager (ex: messages tirés au hasard).

En outre, on s'assurera de la présence de modules de programmation capables de traiter l'information en chaîne de caractères (français). Ainsi, on se demandera s'il existe un analyseur de réponse et jusqu'à quel point cet analyseur est tolérant face aux fautes de français de l'usager, à ses "écarts"de langage, à sa manipulation fautive du clavier (ex: variation majuscules/minuscules, insertion d'espaces, etc.).

Enfin, à l'étape design, on vérifiera si l'on prévoit un examen du français à l'intérieur même du programme (les REMarques par exemple) et de la documentation accompagnant le programme.

RÉSUMÉ

1. Prévoit-on des modules de programmation pour gérer l'interactivité en français?

   oui ◯ non ◯

   Sinon, sont-ils nécessaires? S'ils s'avèrent nécessaires, il faudra donc en prévoir la création. Ex: prévoit-on une banque

de réponses, de commentaires, d'encouragements....

2. Prévoit-on la vérification du respect et de l'utilisation du français aux autres étapes de production: temps nécessaire aux corrections et revisions, raffinement des critères, etc..

oui ◯ non ◯

Sinon, intégrer cette procédure à la démarche.

3. Passer à l'étape codage.

ÉTAPE CODAGE

A l'étape codage on vérifiera trois niveaux:

1. la forme du français à l'affichage et dans la documentation.
2. le niveau du français utilisé.
3. la présentation à l'affichage.

1. Il faudra donc s'assurer que le français écrit et affiché soit grammaticalement correct (sur le plan orthographique, syntaxique, lexical et morphologique)(1).

On surveillera aussi l'homogénéité et la correction de l'ensemble des formules utilisées dans le rapport interactif avec l'usager. Il s'agit ici de vérifier l'application des modules de programmation prévus à l'étape design (voir page 2.1).

De plus, on verra à ce que ne soient pas utilisés des symboles ou abréviations non conformes à l'usage courant (français) ou non compréhensibles par l'étudiant-usager(2).

2. Ces premières vérifications de la forme devront être doublées d'un examen du niveau de langage prévu. Celui-ci devra, évidemment, correspondre au niveau de l'étudiant. On pourra élaborer le processus de vérification à l'étape expérimentation. Le français utilisé devra aussi être proche du langage parlé: interactif, personnalisé, phrases brèves.

---

(1) Y compris les anglicismes!

(2) Voir aussi évaluation du S.I.4.2

De plus, on surveillera la qualité de ce langage interactif:

- est-il trop sec et désincarné?
- est-il trop familier, vulgaire ou insultant?
- utilise-t-il des formules à la mode, donc passagères?
- est-il dénué de formules sexistes, racistes ou discriminatoires?
- est-il intéressant, stimulant, motivant?

3. Enfin, on s'attachera à étudier la présentation des textes français à l'écran. Ici, on se rappelera qu'il est souhaitable de:

- ne pas surcharger l'écran de texte
- présenter les textes en respectant l'opposition minuscule-majuscule.
- ne pas faire dérouler le texte mais le présenter page par page
- standardiser, dans la mesure du possible, la présentation afin de faciliter le repérage des informations: situer les textes dans les mêmes portions d'écran, les questions environ aux mêmes endroits, le titre, etc. Cette standardisation ne devrait pas s'opposer à une certaine diversité dans le style pour rompre tout danger de monotonie.
  Ex: messages d'encouragement variés
- utiliser judicieusement les possibilités graphiques de l'ordinateur utilisé: les soulignés, la couleur, le clignotant pour faire ressortir les éléments de texte les plus importants.

RÉSUMÉ

- Vérifier la correction du français utilisé: élimination des fautes.
- Vérifier la compréhensibilité du texte: ajustement au niveau de l'usager prévu.
- Vérifier la présentation à l'affichage: simple, clair, constant tout au long du programme.

ETAPE EXPÉRIMENTATION

Cette étape va permettre essentiellement de tester la compréhensibilité du texte (affichage et documentation) et de s'assurer du bon fonctionnement de l'interactivité quant aux aspects du français.

Il faudra donc demander:

- est-ce que des mots, expressions ou parties de texte font problème (incompréhension, ambiguïté) pour l'usager(1)? - est-ce que l'usager trouve généralement satisfaisants et faciles à lire les textes présentés?

Il faudra aussi s'assurer de la fonctionnalité du processus interactif:

- l'usager est-il satisfait des instructions, commentaires et réponses fournies? Ex: menus clairs (on ne doute pas du choix qu'on fait)
- l'usager est-il satisfait de la présentation des textes?
- l'usager est-il satisfait(s'il y a lieu) des formats exigés pour entrer de l'information (analyseur de réponse, etc.)?

Enfin, l'étape expérimentation permettra de corriger les dernières fautes et incongruités qui se seraient faufilées jusque là.

---

(1) L'usager pourra être le professeur dans le cas d'une documentation spécifiquement faite pour lui.

RÉSUMÉ

- Existe-t-il des problèmes de compréhension: mauvais niveau de français?

oui ◯ non ◯

- Existe-t-il des problèmes d'interaction homme-machine à cause d'une mauvaise utilisation de la langue?

oui ◯ non ◯

- Existe-t-il encore des fautes?

oui ◯ non ◯

Si l'on répond par un ou plusieurs <u>oui</u> à ces questions, il faut renvoyer le programme à l'auteur accompagné des remarques pertinentes à intégrer.

EVALUATION

SYSTEME INTERNATIONAL D'UNITES (S.I.)

ETAPE DESIGN

1. Vérifier si le programme fait appel à des unités de mesure et/ou à l'écriture de nombres.

   Si oui, voir le point 2. Sinon, l'algorithme ne s'applique pas.

2. Vérifier dans quel système sont rédigées ces unités et/ou écritures.

   Si elles sont dans un système autre que le SI, voir 3.

   Si elles sont en SI, alors voir étape CODAGE

3. Vérifier si on prévoit une sous-routine (module) de transformation en SI, (à l'entrée et/ou à la sortie).

   Si oui, voir CODAGE

   Sinon, passer au point 4.

4. Vérifier dans quelle mesure l'utilisation d'un autre système est justifiée.

Si elle est justifiée, alors l'algorithme ne s'applique pas.

Si elle n'est pas justifiée, signaler à l'auteur la nécessité d'appliquer l'algorithme. Renvoi à 3.

En fait, il faut vérifier si le design prévoit la question SI, donc:

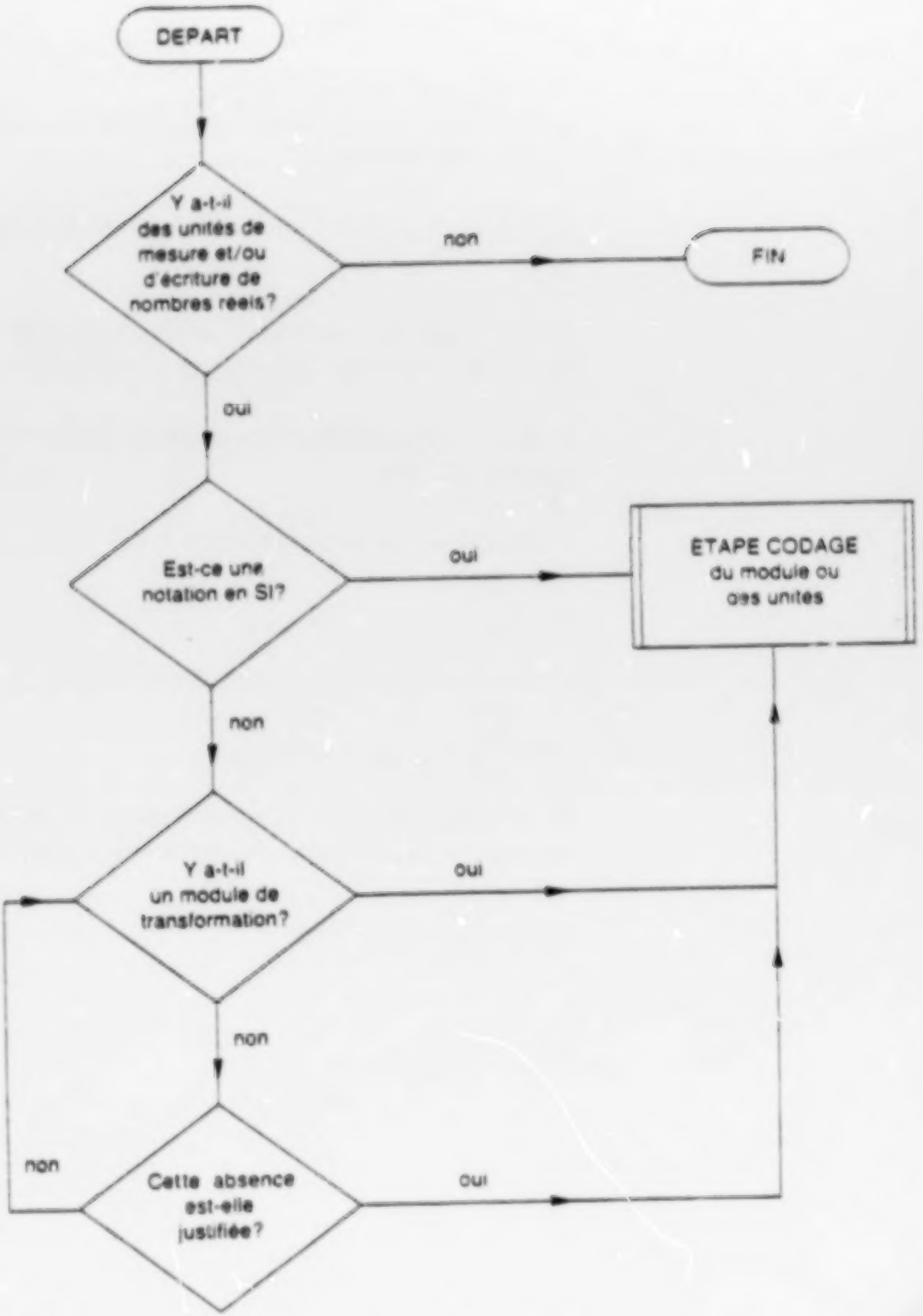
- l'information est-elle sous forme S.I.?

oui ◯ non ◯

- l'information est-elle transformée en S.I.?

oui ◯ non ◯

Si on obtient au moins un non (minimum), il existe une possibilité de problème. On appliquera l'algorithme suivant:


DEPART
Y a-t-il des unités de mesure et/ou d'écriture de nombres reels?
non
FIN
oui
Est-ce une notation en SI?
oui
ETAPE CODAGE du module ou des unités
non
Y a-t-il un module de transformation?
oui
non
non
Cette absence est-elle justifiée?
oui

ÉVALUATION

Système international d'unités

ÉTAPE CODAGE

1. Vérifier - si nécessaire d'après l'étape design - l'existence d'un module de transformation en S.I.

   Si oui, voir 2.

2. Vérifier deux éléments:

   - la justesse scientifique de la transformation voir 3.
   - le format de l'affichage prévu voir 4.

3. Vérifier la transformation. Est-elle scientifiquement pertinente (exactitude des équations utilisées et précision des résultats: 9.10856E-18 n'est pas 0!)? On notera ici une marge possible d'imprécision selon la nature des calculs en cause (objectifs du didacticiel, exigences liées à la matière, etc...)

4. Vérifier l'affichage prévu. Le format prévu à l'affichage est-il conforme aux normes acceptées par le S.I. tant pour l'écriture des nombres et les unités utilisées (y compris les symboles et abréviations. Ex: ou SQRT, 1/3 ou 3.3333...). Ici, il est suggéré d'utiliser la référence: «Système international d'unités» (SI).

Principes d'écriture des unités et des symboles publié par le Bureau de normalisation du Québec (Ministère de l'industrie, du Commerce et du Tourisme) en 1981 sous le numéro BNQ 9990-911.

Dans le cas où il y aurait au moins une réponse négative à l'une des trois questions suivantes il faudra en référer à l'auteur.

- existe-t-il un nécessaire module de transformation en S.I.? oui ◯ non ◯

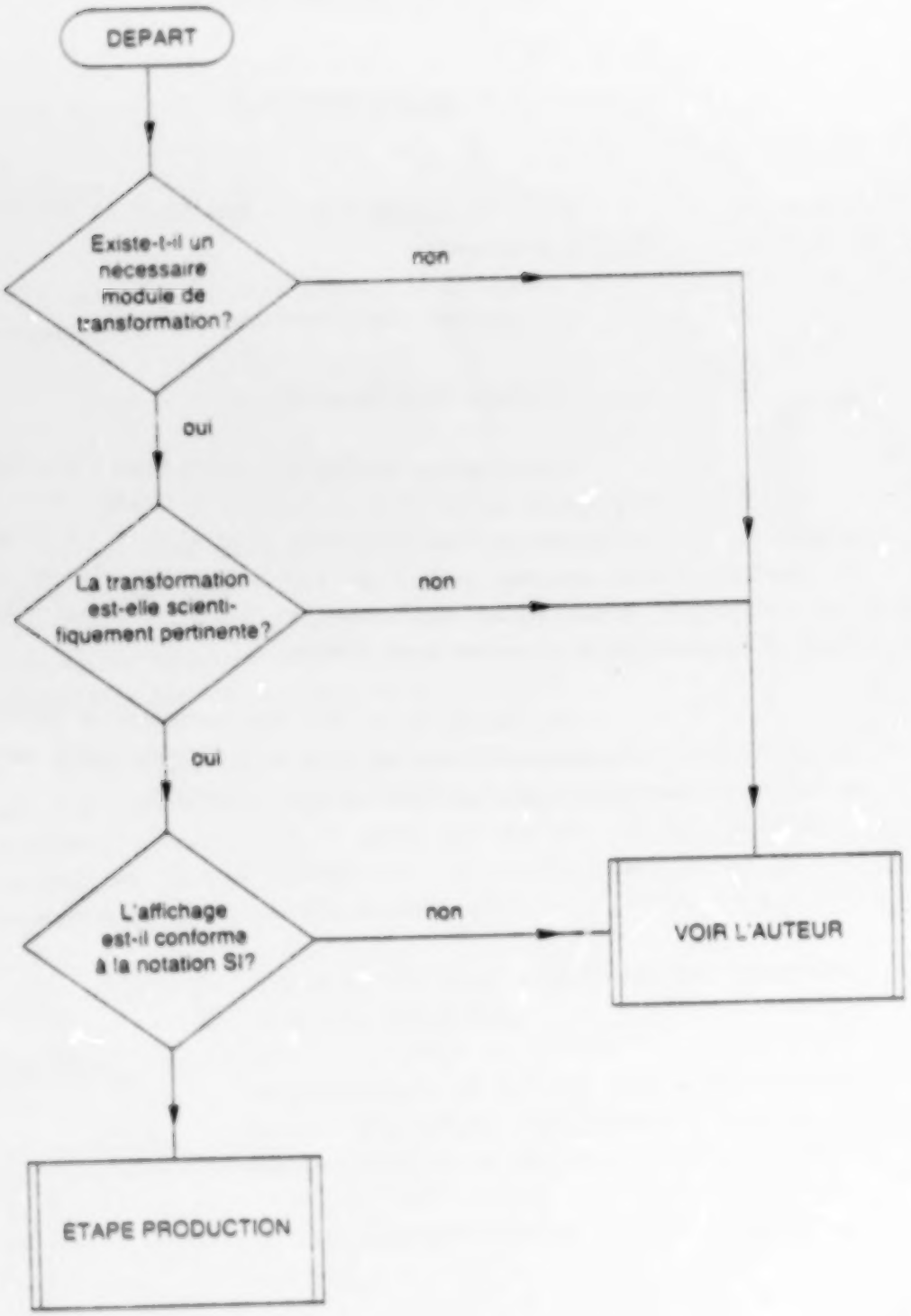
- la transformation est-elle scientifiquement pertinente?

  oui ◯ non ◯

- le format de l'affichage est-il conforme au S.I.?

  oui ◯ non ◯

- Pour systématiser l'approche d'évaluation on pourra appliquer l'algorithme suivant:


DEPART
Existe-t-il un module de transformation nécessaire?
non
oui
La transformation est-elle scientifiquement pertinente?
non
oui
L'affichage est-il conforme à la notation SI?
non
VOIR L'AUTEUR
ETAPE PRODUCTION

SYSTEME INTERNATIONAL D'UNITÉS

ÉTAPE EXPÉRIMENTATION

Ici il faudra vérifier la qualité de la réalisation des étapes précédentes:

1. Validité des transformations produites.

2. Affichage conforme au SI

Par ailleurs, il faudra s'assurer que l'étudiant connait la nature des chiffres et symboles utilisés. Sait-il que le programme utilise le système international? S'il doit fournir des données, sait-il qu'il doit le faire selon ce système? Sinon, sait-il qu'il peut les fournir en un autre système et qu'elles seront transformées?

L'expérimentation doit donc permettre de vérifier le fonctionnement des routines et d'éliminer toute ambiguïté quant à leur utilisation par l'étudiant.

EVALUATION

TRANSFÉRABILITÉ

ÉTAPE DESIGN

A cette étape, on s'assurera d'un maximum de transférabilité en prévoyant:

- la modularité du programme.
- une documentation fouillée.

1. Le programme devra être structuré, clairement organisé en modules distincts réalisant chacun une seule fonction. Les routines "spéciales": graphique, son, calculs scientifiques, entrées-sorties devraient se démarquer spécifiquement du reste du programme. Au total, on examinera donc le découpage du programme.

2. On vérifiera aussi la documentation prévue. En prévoit-on? Si oui, quels éléments prévoit-on inscrire quant à la transférabilité du programme? A-t-on pensé à cette possibilité? Si oui, quelles informations va-t-on offrir? Il faudra donc vérifier si la documentation va fournir des données sur:

: le support sur lequel le prugramme est enregistré (cassette, disquette...).

: la configuration du matériel nécessaire pour faire tourner le programme (quelle machine, quantité de mémoire, types de périphériques nécessaires, etc.).

: le langage de programmation utilisé (y compris la version).

Il sera donc utile d'examiner l'information prévue quant à l'environnement technique.

3. Prévoit-on disposer de routines toutes faites ou de faire des routines pour isoler les fonctions non transférables. Toutes les fonctions d'entrées-sorties devraient ainsi être isolées car elles sont généralement propres à chaque équipement.

RESUME

- Le programme est-il structuré en modules?

oui ◯ non ◯

- Prévoit-on une documentation précise sur l'environnement technique?

oui ◯ non ◯

Si ces deux critères sont respectés, il y aura passage à l'étape codage.

ETAPE CODAGE

Ici, il faudra d'abord vérifier le respect de la structure du programme élaboré à l'étape précédente. Les modules existent-ils tel que prévu? Dans un rapport et une structure respectueuse d'une programmation "structurée"?

Mais au-delà de cet aspect général, il faudra s'assurer d'une bonne documentation interne du programme(1). Cette documentation devrait porter sur:

- une identification - en début de programme - du langage utilisé (y compris le numéro de version).
- des remarques qualifiant chaque module.
- des remarques signalant l'utilisation de particularités propres au langage utilisé et n'ayant que peu de chance d'être retrouvées ailleurs.
- des remarques précédant les modules faisant appel à des particularités propres au matériel de base (ex: graphiques, niveau de résolution, etc.).
- une documentation «au petit point» accompagnant les passages spéciaux, les tours de «passe-passe» et les «créations» du programmeur.

Enfin, la version finale du programme devra être enregistrée en fichier ASCII afin de favoriser sa transportabilité.

---

(1) Dans le cas où une telle documentation ralentirait la vitesse de déroulement ou augmenterait indûment le volume du programme, il faudra produire deux versions: l'une documentée, l'autre expurgée, prête à fonctionner.

RESUME

- Le programme est-il codé selon les règles de la programmation «structurée» (modulaire)?

  oui ◯ non ◯

- Le programme est-il documenté intérieurement?

  oui ◯ non ◯

- Le programme est-il codé en ASCII?

  oui ◯ non ◯

Si une ou plusieurs de ces conditions ne sont pas respectées, il faut reviser l'étape codage. Après, on pourra - si nécessaire - passer à l'étape expérimentation.

ETAPE EXPERIMENTATION

Ici, il va de soi que l'expérimentation consiste à transférer le programme dans un autre environnement (autre langage et/ou autre matériel).

Dans la mesure où les exigences antérieures auront été respectées, ce transfert sera facilité.

Un certain nombre de difficultés peuvent pourtant apparaître. Difficultés qui peuvent d'ailleurs s'avérer insurmontables.

Avant de procéder au transfert, il faudra donc examiner un certain nombre de facteurs susceptibles de nuire à la transférabilité:

- sur le plan du matériel, il faut s'assurer de la possibilité du transfert: acceptabilité du support, problème de reformattage (nombre de lignes et colonnes), problème de la résolution graphique, des correspondances sonores...
- sur le plan du langage de programmation, on surveillera la possibilité de reproduire les diverses fonctions et opérations d'un langage dans un autre sans perte de fiabilité ou de contrôle (ex: fonctions graphiques...).

Si de telles assurances peuvent être acquises, on pourra alors procéder au transfert.

NOTES DE DISCUSSION

	oui	non	N.S.P.*	commentaires
ETAPE DESIGN				
INTERACTION				
- Le programme répond t-il à des critères efficaces d'interaction (modularité, simplicité, interactivité, souplesse)?				
- Le programme permet-il le maximum de contrôle par l'étudiant?				
FRANÇAIS				
- Prévoit-on des modules de programmation pour gérer l'interactivité en français?				
- Prévoit-on la vérification du respect et de l'utilisation du français aux autres étapes de production?				

* N.S.P. = ne s'applique pas

	oui	non	N.S.P.	commentaires
SI				
- L'information donnée est-elle sous forme S.I.?				
- L'information sera-t-elle sous forme S.I.?				
TRANSFÉRABILITÉ				
- Le programme est-il structuré en modules?				
- Prévoit-on une documentation précise sur l'environnement technique?				

ÉTAPE CODAGE	oui	non	N.S.P.	commentaires
INTERACTION				
- Le codage respecte-t-il le design établi (modules, affichage...)?				
- Le codage assure-t-il une bonne documentation du programme pour l'utilisateur?				
- Le codage assure-t-il un maximum de facilité d'utilisation pour l'étudiant?				
- Le codage assure-t-il une bonne esthétique de chaque page-écran?				
- Le codage protège-t-il le programme au maximum?				
FRANÇAIS				
- Le français est-il correct (fautes, etc.)?				
- Le français est-il compréhensible (niveau de l'usager)?				

	oui	non	N.S.P.	commentaires
- Le français est-il bien présenté (texte)?				
SI				
- Existe-t-il un nécessaire module de transformation en S.I.?				
- La transformation est-elle scientifiquement pertinente?				
- Le format de l'affichage est-il conforme au S.I.?				
TRANSFÉRABILITÉ				
- Le programme est-il codé selon les règles de la programmation structurée?				
- Le programme est-il documenté intérieurement?				
- Le programme est-il codé en ASCII?				
- A-t-on isolé les fonctions non-transférables dans des modules distincts?				

ETAPE EXPÉRIMENTATION	oui	non	N.S.P.	commentaires
INTERACTION				
- Existe-t-il des problèmes sérieux et/ou systématiques dans le rapport étudiant-programme?				
FRANÇAIS				
- Existe-t-il des problèmes de compréhension (niveau de français inapproprié didactiquement)?				
- Existe-t-il encore des fautes?				
SI				
- L'affichage est-il conforme aux normes SI?				
TRANSFÉRABILITÉ				
- Le transfert est-il possible sur le plan technique?				
- Le transfert est-il possible sur le plan du langage de programmation?				